Migros Cross-Math Solver

Pascal Lüscher

January 14, 2020

1 Problem

The Cross-Math puzzle is a puzzle with 9 numbers to fill. The rules are simple:

- Mathematical operation ordering does not count, simply calc from left to right and from top to bottom
- Each cell is filled with a number between 1 - 9
- Each number can only appear once

The puzzle presented is the following:

	$-$		$*$		$=$	3
$-$		$+$		$*$		
	$+$		$*$		$=$	20
$+$		$/$		$*$		
	$+$		$+$		$=$	20
$=$		$=$		$=$		
13		1		20		

2 QCP Mathematical formulation

In this section i present some quadratic constraint programming solutions

2.1 With Big-M

$$
\begin{aligned}
\min \quad & 0 && && (1)\\
\text{s.t.} \quad & x_i && \geq 1, && \forall i \in \{1,\ldots,9\} \quad (2)\\
& x_i && \leq 9, && \forall i \in \{1,\ldots,9\} \quad (3)\\
& x_i - x_j && \leq -\epsilon + y_{i,j}M, && \forall i,j \in \{1,\ldots,9\} \quad (4)\\
& x_i - x_j && \geq \epsilon - (1-y_{i,j})M, && \forall i,j \in \{1,\ldots,9\} \quad (5)\\
& (x_0 - x_1)x_2 && = 3 && (6)\\
& (x_3 + x_4)x_5 && = 20 && (7)\\
& x_6 + x_7 + x_8 && = 20 && (8)\\
& x_0 - x_3 + x_6 && = 13 && (9)\\
& \frac{(x_1 + x_4)}{x_7}, && = 1 && (10)\\
& x_2 x_5 x_8 && = 20 && (11)\\
& x_i && \in \mathbb{N}, && \forall i \in \{1,\ldots,9\} \quad (12)\\
& y_{i,j} && \in \{0,1\}, && \forall i,j \in \{1,\ldots,9\} \quad (13)\\
& && && (14)
\end{aligned}
$$

M is very big and ϵ is a very small number.

Equation 2 – Equation 5 are the basic requirements for the corss-math puzzle. Equation 6 – Equation 11 are the specific equations for this problem.

2.2 One Hot Matrix

The idea behind a one hot matrix is to add a binary variable for each possibility. In our case this means to add 9 binary variables for every field.

$$
\begin{aligned}
\min \quad & 0 && && \qquad (15)\\
\text{s.t.} \quad & \sum_{j=1}^{j\leq 9} x_{i,j} &&= 1, && \forall i \in \{1,\ldots,9\} \qquad (16)\\
& \sum_{i=1}^{i\leq 9} x_{i,j} &&= 1, && \forall j \in \{1,\ldots,9\} \qquad (17)\\
& y_i &&= \sum_{j=1}^{j\leq 9} j x_{i,j}, && \forall i \in \{1,\ldots,9\} \qquad (18)\\
& (y_0 - y_1)y_2 &&= 3 && \qquad (19)\\
& (y_3 + y_4)y_5 &&= 20 && \qquad (20)\\
& y_6 + y_7 + y_8 &&= 20 && \qquad (21)\\
& y_0 - y_3 + y_6 &&= 13 && \qquad (22)\\
& \frac{(y_1 + y_4)}{y_7} &&= 1 && \qquad (23)\\
& y_2 y_5 y_8 &&= 20 && \qquad (24)\\
& x_{i,j} &&\in \{0,1\}, && \forall i,j \in \{1,\ldots,9\} \qquad (25)\\
& && && \qquad (26)
\end{aligned}
$$

The constraints Equation 16 and Equation 17 define the uniqueness of a number in the problem. With Equation 18 a linear expression y_i is defined that expresses the actual value of the cell i. This linear expression is then used in Equation 19 – Equation 24 to define the actual problem. These constraints are the same as Equation 6 – Equation 11.

3 Backtracking

The puzzle is small and can easily be solved using a naive backtracking algorithm. A pseudo code algorithm is shown in Listing 1. The function checkAllPredicates checks if all conditions are met.

Listing 1: Backtrack algorithm

```
int[] field = new int[9];
Func<int,int,int,bool> predicates;
Solve(int currentField, int[] availableNumbers)
  if (!checkAllPredicates) return false;
  if (currentField > 8) return true

  foreach(number in availableNumbers) do
    field[currentField] = number
      if (Solve(currentField + 1, availableNumbers without number))
        return true
  field[currentField] = 0
  return false
```

4 Solution

9	−	6	*	1	=	3
−		+		*		
3	+	2	*	4	=	20
+		/		*		
7	+	8	+	5	=	20
=		=		=		
13		1		20		